



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/737,527	12/13/2000	Paul F. Austin	5150-47600	9833

35690 7590 10/11/2006

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.  
700 LAVACA, SUITE 800  
AUSTIN, TX 78701

EXAMINER

BASOM, BLAINE T

ART UNIT PAPER NUMBER

2173

DATE MAILED: 10/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No. 09/737,527	Applicant(s) AUSTIN ET AL.	
	Examiner Blaine Basom	Art Unit 2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 10 July 2006.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 89-93,95-98,100-115 and 117-127 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 89-93,95-98,100-115 and 117-127 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## **DETAILED ACTION**

### ***Response to Arguments***

The Examiner acknowledges the Applicants' amendments to claims 89-92, 95-98, 101-102, 104-106, 108-109, 111-115, 118, and 121, the Applicants' cancellation of claim 94, and the Applicants' addition of new claims 124-127.

Regarding independent claims 89 and 105, the Applicants argue that Austin (U.S. Patent No. 6,370,569) and Semenzato (U.S. Patent No. 5,903,728), presented in the previous Office Action, fail to teach automatically displaying a second GUI element in a graphical program in response to determining that a first GUI element cannot display data of a first data type, as is claimed. The Examiner, however, respectfully disagrees with this argument. As is further described below, Semenzato teaches automatically displaying a second GUI element (i.e. for a browser plug-in) in response to determining that a first GUI element (i.e. for a browser) cannot display data of a first data type. Austin further teaches displaying such GUI elements within a graphical program for receiving data from a remote data source, as is shown below. Accordingly, Austin and Semenzato in fact teach automatically displaying a second GUI element in a graphical program in response to determining that a first GUI element cannot display data of a first data type, as is claimed.

Concerning independent claims 106 and 122, the Applicants argue that Austin and Semenzato fail to teach automatically displaying information in a graphical program to indicate an invalid condition if a first GUI element cannot display data of a first data type. The Examiner respectfully disagrees with this argument. First, the Examiner respectfully notes that the claim language in question, i.e. "displaying information in the graphical program to indicate an invalid

Art Unit: 2173

condition if the first GUI element cannot display data of the first data type,” does not limit the claim scope, because it does not require steps to be performed. The displaying step is not necessarily performed, since it is only performed if the first GUI element cannot display data of the first data type. That is, given the broadest, most reasonable interpretation of claims 106 and 122 – when the GUI element can display data of the first data type – an invalid condition is not displayed. Accordingly, even if Austin and Semenzato failed to teach displaying an invalid condition, they would still be sufficient to read on claims 106 and 122. Nevertheless, the Examiner respectfully notes that Austin in fact teaches that if a first GUI element cannot recognize the first data type, and there is no associated plug-in for the first data type, then an invalid condition is displayed to the user (for example, see column 14, lines 50-59; and column 18, lines 3-11).

As per new claims 124-127, the Applicants generally argue that Austin and Semenzato do not teach the subject matter claimed. The Examiner respectfully disagrees. As is shown below in the rejections for claims 124-127, Austin and Semenzato in fact teach the subject matter of these new claims.

The Applicants’ arguments have thus been fully considered, but are not persuasive.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claim 123 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 23 expresses that the method steps of claim 89 are performed during creation of a graphical program “without executing the graphical program.” Such a limitation, however, is not described or suggested in the specification. In fact, the specification appears to teach the contrary, i.e. that the method steps of claim 89 are performed during execution of the graphical program (see e.g. page 7, lines 3-7; and page 10, line 26 – page 11, line 5).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 89-93, 95-98, 100-115, and 117-127 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,370,569, which is attributed to Austin, and also over U.S.

Art Unit: 2173

Patent No. 5,903,728, which is attributed to Semenzato. In general, Austin teaches a DataSocket client, which may be implemented within a plurality of different types of applications, to access data from various sources that provide data of various formats (for example, see column 2, lines 20-42).

Specifically regarding claims 89, 105, 106, and 122, Austin teaches: displaying a GUI element for the DataSocket, which can be in graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate the functionality of the graphical program (for example; see figures 6-7; column 16, lines 1-44; and column 17, lines 34-44); receiving user input (e.g. a URL) specifying a data source with which to associate the GUI element (for example, see column 2, line 62 – column 3, line 21; and column 12, line 55 – column 12, line 6); in response to receiving the user input, automatically configuring the GUI element to receive data from the specified data source (for example, see column 3, lines 9-21; and column 13, lines 50-65); receiving data from the specified data source, wherein the data includes information specifying a first type of data (see column 3, lines 9-33; and column 13, line 66 – column 14, line 9); automatically determining if there is built-in support for the specified data type, i.e. that the GUI element can process and display the data (for example, see column 14, lines 9-59); and displaying the received data content from the specified data source in the first GUI element (for example, see column 16, lines 1-44). Austin further teaches that if the DataSocket cannot recognize the data type of the data from the data source, than in response, it attempts to find a plug-in to process the data (for example, see column 3, lines 21-33). Austin, however, does not explicitly teach: automatically displaying a second GUI element in the graphical program in response to determining that the

Art Unit: 2173

first GUI element cannot display data of the first type, wherein the second GUI element can display data of the first type; and displaying the received data from the specified data source on the second GUI element, as is expressed in claim 89. Nevertheless, such teachings are well-known in the art.

For example, Semenzato discloses a browser application, which like the DataSocket of Austin, is used to access data from remote sources specified by URLs. As known in the art, plug-ins associated with such browsers are instantiated in response to the browser application receiving data from a user-specified source, wherein the data comprises a type of data in which the browser cannot process. Semenzato specifies that such plug-ins may create particular GUI elements for displaying the data (for example, see column 3, lines 23-35). Additionally, it is understood that such GUI elements are often displayed in their own window, separate from that of the browser, as is known in the art. In such circumstances, a first GUI element, particularly that associated with a browser, is displayed on a display of a first computer system; user input specifying a data source with which to associated the first GUI element is received; it is automatically determined that the first GUI element cannot display data of the first data type; a second GUI element, specifically that associated with a plug-in, is automatically displayed in response to determining that the first GUI element cannot display data of the first data type, wherein the second GUI element can display data of the first type; and the received data from the specified data source is displayed on the second GUI element.

Consequently, it would have been obvious to one of ordinary skill in the art, having the teachings of Austin and Semenzato before him at the time the invention was made, to modify the plug-ins taught by Austin to include their own GUI, like taught by Semenzato, whereby if the

Art Unit: 2173

first GUI element cannot display the received data, the GUI of the plug-in does. It would have been advantageous to one of ordinary skill to utilize such a combination because such plug-in GUIs allows customized interface elements for each of a plurality of data types, thus allowing for more specific control elements to be displayed for a given data type, as is demonstrated by Semenzato. Consequently, Austin and Semenzato are considered to teach a method like that recited in claim 89. It is understood that this method is implemented with a computer (for example, see column 7, line 41 – column 8, line 6). Such a computer implementing the above-described method of Austin and Semenzato is considered to comprise a memory medium, like that recited in claim 105, which stores program instructions for configuring a GUI element to subscribe to a data source. Additionally, and specifically regarding claim 106, Austin discloses that if the DataSocket cannot recognize the first data type, and there is no associated plug-in for the first data type, then an invalid condition is indicated (for example, see column 14, lines 50-59). It is apparent that such an invalid condition would be communicated, e.g. displayed, to the user. Accordingly, Austin and Semenzato are further considered to teach a method like that recited in claim 106. It is understood that this method is implemented with a computer, as is asserted above. Such a computer implementing the above-described method of Semenzato is considered to comprise a memory medium, like that recited in claim 122, which stores program instructions for configuring a GUI element to subscribe to a data source.

As per claims 96-98, 113-115, and 123, Austin teaches that method described above can be implemented during development of a graphical program, the graphical program comprising a user interface including the DataSocket GUI element and a block diagram (for example, see column 7, lines 48-55; FIGS. 11 and 12; and column 17, lines 34-59). The GUI element



Art Unit: 2173

associated with the DataSocket is thus displayed in a user interface of this graphical program. Accordingly, displaying the first GUI element, receiving the user input specifying the data source, automatically configuring the first GUI element, receiving data, automatically determining, automatically substituting the second GUI element, and displaying the received data can be performed during development of the graphical program, prior to executing the completed graphical program. Concerning claims 97, 98, 114, and 115, it is understood that, during execution, either of the plug-in or browser is operable to receive and display data from the specified data source, as is described above.

Concerning claims 90 and 107, it is understood that the data source accessed by the DataSocket can be located remotely from the computer system executing the DataSocket, and coupled to the computer system over a network, wherein the data source is specified by a URL input by the user (for example, see column 2, line 63- column 3, line 21; and column 8, line 63 – column 9, line 8 of Austin). Consequently, it is further understood that configuring the first GUI element, namely that associated with the DataSocket, comprises automatically configuring the element to connect to the data source in order to receive data from the source.

As per claims 91-93 and 108-111, it is understood that the only user input involved in configuring the DataSocket to receive and display data from a data source may be that of entering a URL specifying the data source (for example, see column 2, line 63 – column 3, line 21 of Austin). The first GUI element, i.e. that associated with the DataSocket, is thus automatically configured without user programming and without user input specifying source code. It is understood that, if the data is of an appropriate type, this GUI element within the graphical program receives and displays data from the specified data source after it is configured

Art Unit: 2173

to do so, like expressed in claim 111. Additionally, Austin discloses that the URL is entered via a text box, i.e. a dialog box, like recited in claim 93.

With respect to claims 95 and 112, it is understood that the data source accessed by the DataSocket of Austin may comprised in a second computer system, namely a server, which is remotely located from the computer system executing the DataSocket, and wherein the computer system is operable to connect to the second computer system over a network (for example, see column 2, line 63- column 3, line 21; and column 8, line 63 – column 9, line 8 of Austin). Consequently, it is understood that configuring the DataSocket to receive data from a specified source comprises automatically configuring the DataSocket to connect to a second computer system.

Regarding claims 100-101 and 117-118, it is understood that the specified data source accessed by the DataSocket of Semenzato may be an HTTP server (for example, see column 2, line 63- column 3, line 21; and column 8, line 63 – column 9, line 8 of Austin). This data source may thus be remotely located from the computer system of the user, and therefore, the first data source may be a remote data source associated with a remote computer, wherein automatically configuring the GUI element comprises automatically configuring the GUI element to connect to the remote data source and receive data from the remote data source during execution of the graphical program.

In reference to claims 102 and 119, Austin further teaches executing a computer program operable to publish live data to the remote data source, whereby the GUI element associated with the DataSocket is operable to display the live data (for example, see column 12, lines 11-27).

In reference to claims 103-104 and 120-121, Austin teaches a method like that of claims 89 and 106, whereby as described above, a GUI element of a plug-in is automatically configured to connect to a specified remote data source and receive and display data from the remote data source. It is understood that the data may be live data, like recited in claims 103 and 120, or live measurement data, like recited in claims 104 and 121 (for example, see column 6, lines 43-48; and column 12, lines 11-27).

As per claim 124, Austin teaches configuring a first GUI element in a graphical program to receive data from a data source, and when the first GUI element cannot recognize data received from the source, Austin teaches executing a plug-in to process the data, as is described above. As is further described above, Semenzato teaches that such plug-ins may comprise their own GUI elements, i.e. a second GUI element, to process and display the data. It is understood that such GUI elements are often displayed in their own window, separate from that of the browser, which is displayed over the browser window or instead of the browser window, as is known in the art. In such cases, the second GUI element is automatically substituted for the first GUI element, wherein substituting the second GUI element for the first GUI element comprises automatically ceasing display of the first GUI element in the graphical program and automatically displaying the second GUI element in the graphical program instead of the first GUI element.

Concerning claims 125 and 126, Austin teaches that the graphical program includes a user interface portion and a block diagram portion, wherein the block diagram portion includes a plurality of interconnected nodes (see e.g. FIG. 12), and wherein displaying the first GUI

Art Unit: 2173

element in the graphical program comprises displaying the first GUI element in the user interface portion of the graphical program (see e.g. FIG. 11).

As per claim 127, Austin discloses that if the DataSocket, i.e. first GUI element, cannot recognize the first data type, and there is no associated plug-in for the first data type, then an invalid condition is indicated (for example, see column 14, lines 50-59). It is apparent that such an invalid condition would be communicated, e.g. displayed, to the user proximally to the first GUI element in the graphical program (see e.g. column 18, lines 3-11).

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2173

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blaine Basom whose telephone number is (571) 272-4044. The examiner can normally be reached on Monday through Friday, from 8:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (571) 272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

**TADESSE HAILU**  
*Patent Examiner*



btb  
9/30/2006